Отчёт по лабораторной работе 11

Настройка безопасного удалённого доступа по протоколу SSH

Авдадаев Джамал Геланиевич

Содержание

Список иллюстраций

Список таблиц

1 Введение

1.1 Цель работы

Приобретение практических навыков по настройке удалённого доступа к серверу с помощью SSH.

2 Процесс работы

2.1 Запрет удалённого доступа по SSH для пользователя root

2.1.1 Попытка подключения по SSH под пользователем root

С клиента была выполнена попытка подключения к серверу через SSH под пользователем *root*.
SSH запросил подтверждение ключа сервера, после чего несколько попыток ввода пароля завершились неудачно.
Аутентификация была отклонена — вход под root не был разрешён.

Рис. 2.1: Неуспешная попытка подключения под root

2.1.2 Изменение конфигурации SSH для запрета root-доступа

На сервере был открыт файл конфигурации `/etc/ssh/sshd_config`, где параметр:

```
PermitRootLogin no
```

запрещает удалённый вход для пользователя root.

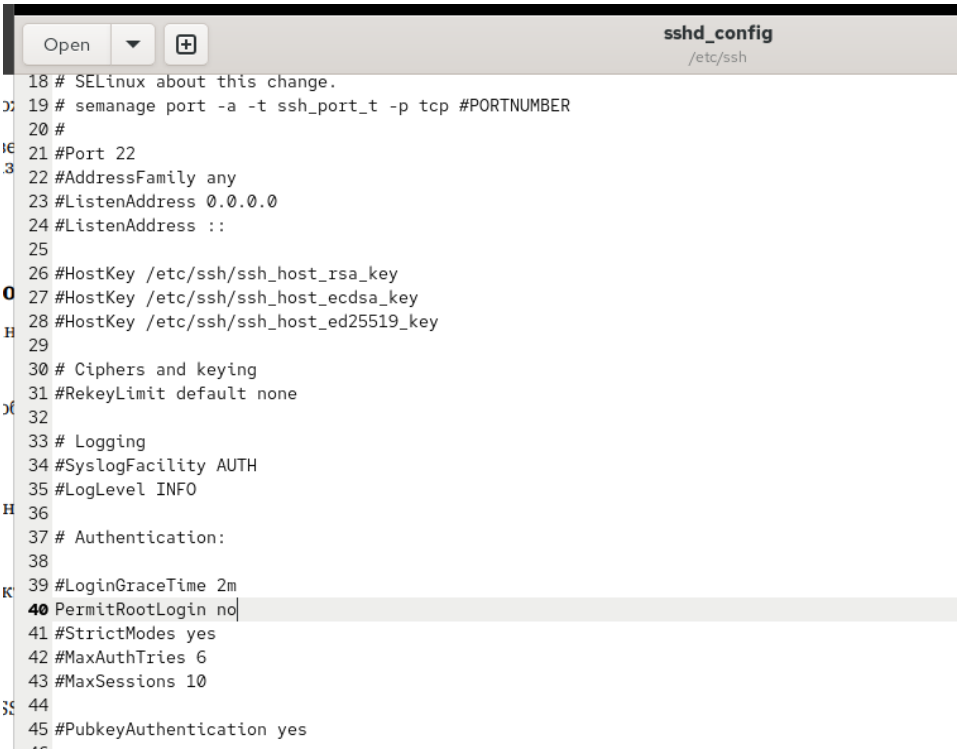

Рис. 2.2: Файл sshd_config — запрет root

После выполнения перезапуска службы SSH изменения вступили в силу.

2.1.3 Повторная попытка подключения под root

После перезапуска SSH-сервера повторная попытка подключения под root вновь завершается отказом.

Теперь это ожидаемое поведение, так как вход root официально запрещён в конфигурации SSH.

Рис. 2.3: Отказ в доступе после запрета PermitRootLogin

2.2 Ограничение списка пользователей для удалённого доступа по SSH

2.2.1 Попытка подключения под обычным пользователем

Пользователь *dgavdadaev* успешно подключился по SSH к серверу. Аутентификация выполнена успешно, доступ предоставлен.

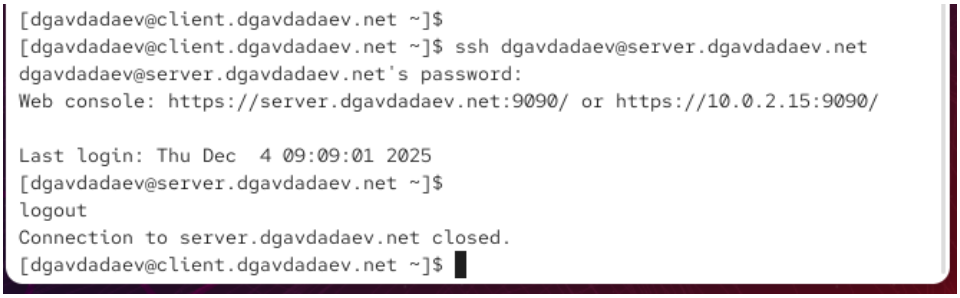

Рис. 2.4: Успешный вход пользователя dgavdadaev

2.2.2 Добавление ограничения AllowUsers

В файл `/etc/ssh/sshd_config` добавлена строка:

```
AllowUsers vagrant
```

Это ограничивает вход только для пользователя *vagrant*.

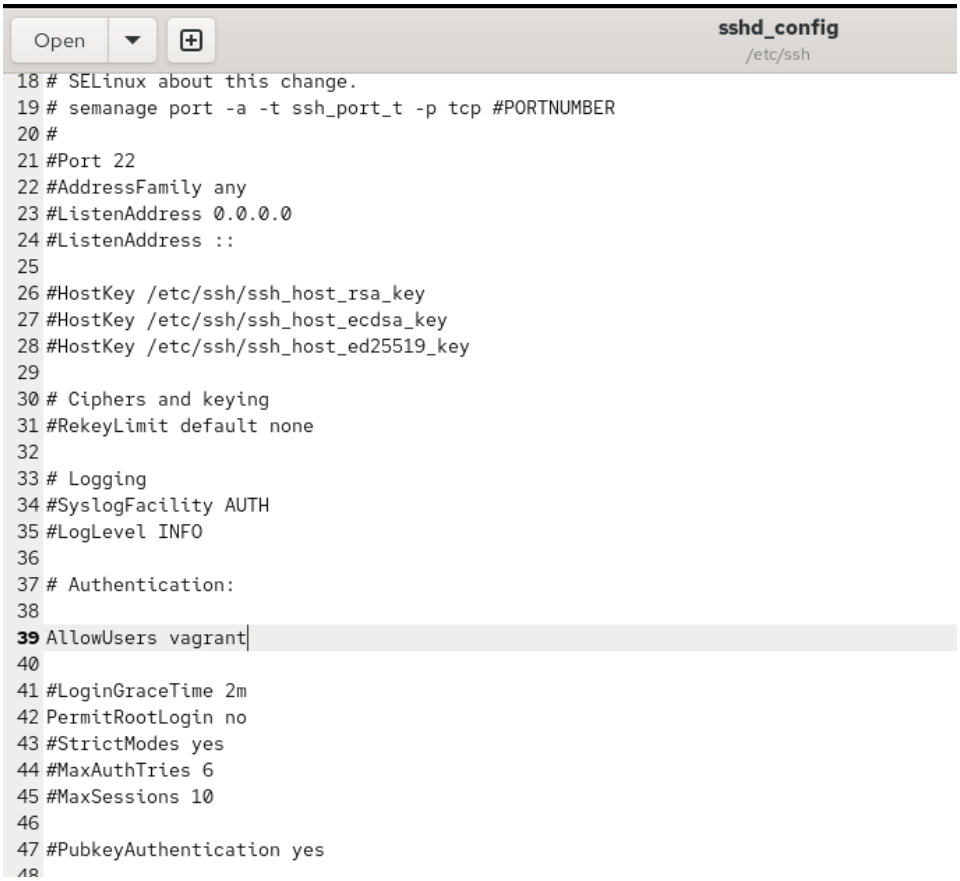

Рис. 2.5: Файл sshd_config — AllowUsers vagrant

После перезапуска sshd изменения вступили в силу.

2.2.3 Повторная попытка подключения под пользователем dgavdadaev

После добавления ограничения пользователь *dgavdadaev* больше не может войти в систему.

Попытки входа завершаются сообщением об отказе в доступе.

```
[dgavdadaev@client.dgavdadaev.net ~]$
[dgavdadaev@client.dgavdadaev.net ~]$ ssh dgavdadaev@server.dgavdadaev.net
dgavdadaev@server.dgavdadaev.net's password:
Permission denied, please try again.
dgavdadaev@server.dgavdadaev.net's password:
Permission denied, please try again.
dgavdadaev@server.dgavdadaev.net's password:
dgavdadaev@server.dgavdadaev.net: Permission denied (publickey,gssapi-keyex,gssa
pi-with-mic,password).
[dgavdadaev@client.dgavdadaev.net ~]$
```

Рис. 2.6: Отказ из-за AllowUsers

2.2.4 Расширение списка AllowUsers

В конфигурацию было добавлено расширение списка разрешённых пользователей:

```
AllowUsers vagrant dgavdadaev
```

Open

sshd_config
/etc/ssh

```
17 # If you want to change the port on a SELinux system, you have to tell
18 # SELinux about this change.
19 # semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
20 #
21 #Port 22
22 #AddressFamily any
23 #ListenAddress 0.0.0.0
24 #ListenAddress ::
25
26 #HostKey /etc/ssh/ssh_host_rsa_key
27 #HostKey /etc/ssh/ssh_host_ecdsa_key
28 #HostKey /etc/ssh/ssh_host_ed25519_key
29
30 # Ciphers and keying
31 #RekeyLimit default none
32
33 # Logging
34 #SyslogFacility AUTH
35 #LogLevel INFO
36
37 # Authentication:
38
39 AllowUsers vagrant dgavdadaev
40
41 #LoginGraceTime 2m
42 PermitRootLogin no
43 #StrictModes yes
44 #MaxAuthTries 6
45 #MaxSessions 10
46
47 #PubkeyAuthentication yes
```

Рис. 2.7: Файл sshd_config — добавление второго пользователя

После перезапуска sshd пользователь вновь получил возможность подключаться.

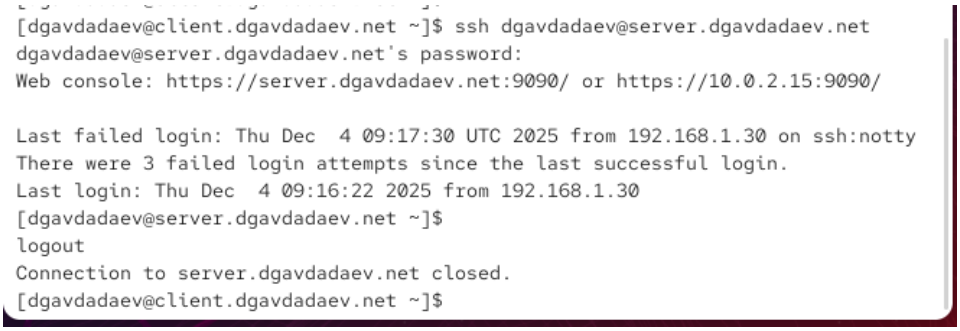

Рис. 2.8: Успешный вход после добавления пользователя в AllowUsers

2.3 Настройка дополнительных портов для удалённого доступа по SSH

2.3.1 Добавление второго SSH-порта

В конфигурационном файле `/etc/ssh/sshd_config` была найдена строка `Port` и ниже добавлены два порта:

```
Port 22
Port 2022
```

Это позволяет одновременно слушать стандартный порт SSH и дополнительный, чтобы избежать полной потери доступа при ошибке в настройках.

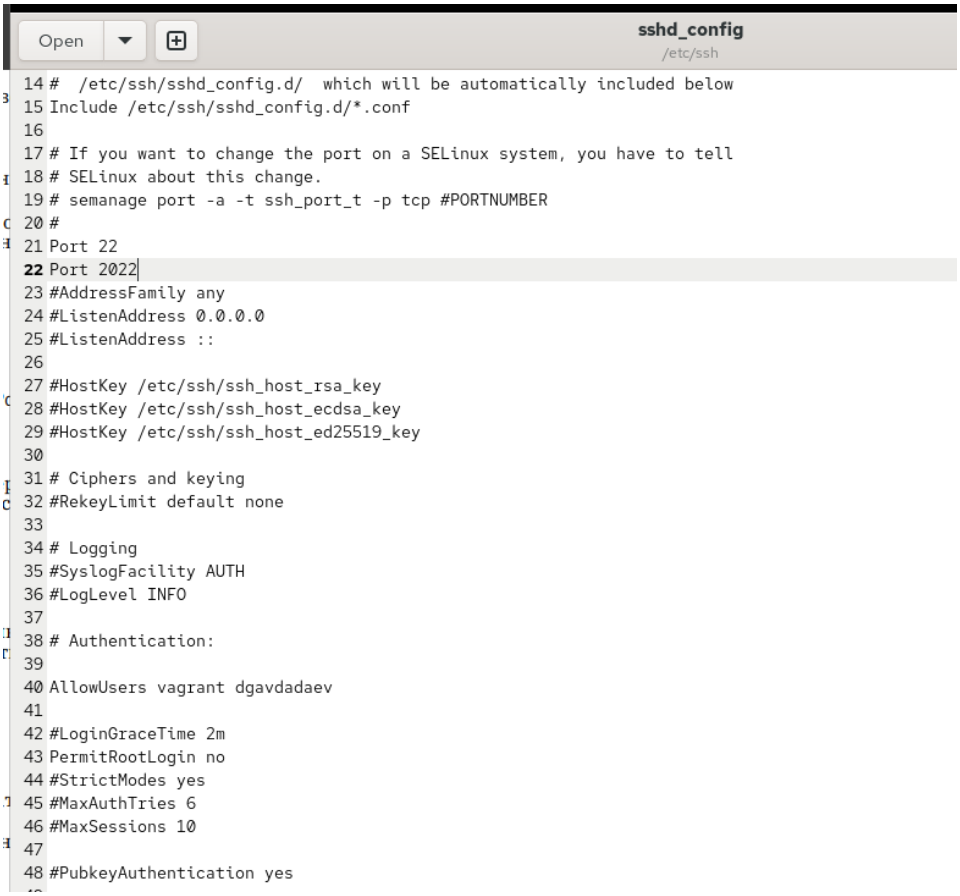

Рис. 2.9: Добавление порта 2022

После сохранения изменений служба SSH была перезапущена.

2.3.2 Ошибка SELinux при запуске sshd

После перезапуска расширенный статус `systemctl status sshd` показал ошибку:

sshd не смог привязаться к порту 2022, так как SELinux запрещал использование этого порта для сервиса ssh.

Системные сообщения:

- *Bind to port 2022 failed: Permission denied*
- SELinux блокирует назначение нестандартного порта SSH.

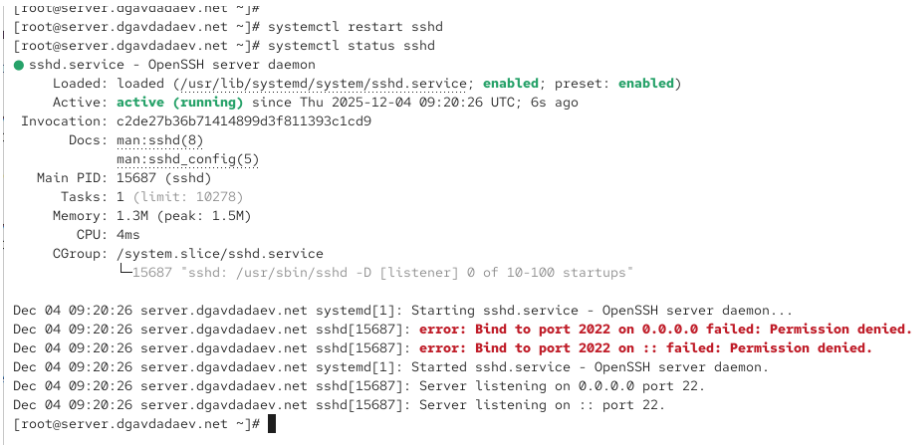

Рис. 2.10: Ошибка привязки к порту 2022

2.3.3 Исправление SELinux и настройка firewall

Для разрешения использования порта 2022 был выполнен ввод метки SELinux:

```
semanage port -a -t ssh_port_t -p tcp 2022
```

Затем в firewall открыт новый порт:

```
firewall-cmd --add-port=2022/tcp
firewall-cmd --add-port=2022/tcp --permanent
```

После перезапуска sshd статус показал, что сервер теперь прослушивает оба порта — 22 и 2022.

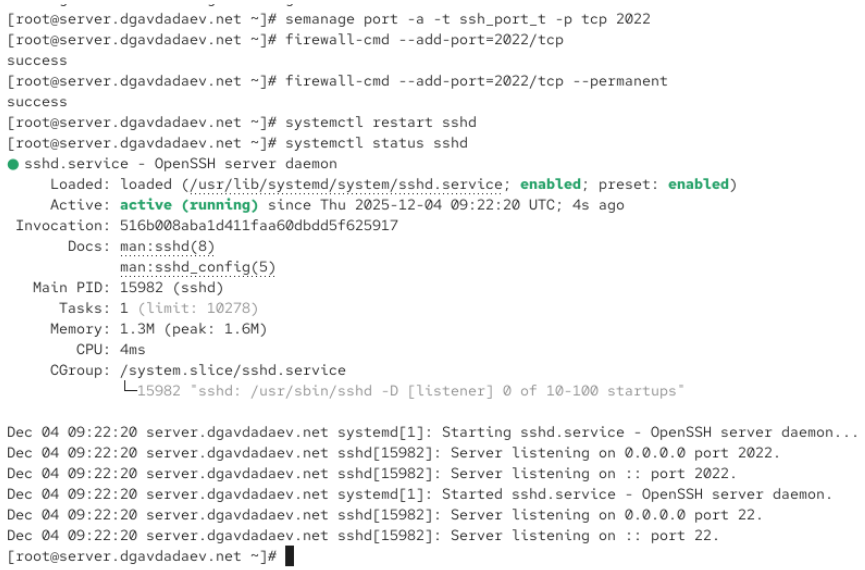

Рис. 2.11: Успешное добавление порта SELinux и firewall

2.3.4 Проверка подключения через порт 22

Была выполнена команда подключения:

`ssh [dgavdadaev@server.dgavdadaev.net](mailto:dgavdadaev@server.dgavdadaev.net)`

Вход выполнен успешно, а затем через `sudo -i` получен доступ root.
После выхода из обоих сеансов соединение закрыто.

2.3.5 Подключение через порт 2022

Далее была выполнена проверка альтернативного порта:

`ssh -p2022 [dgavdadaev@server.dgavdadaev.net](mailto:dgavdadaev@server.dgavdadaev.net)`

Вход снова прошёл успешно.
Поведение идентично предыдущему — SSH корректно работает через новый порт.

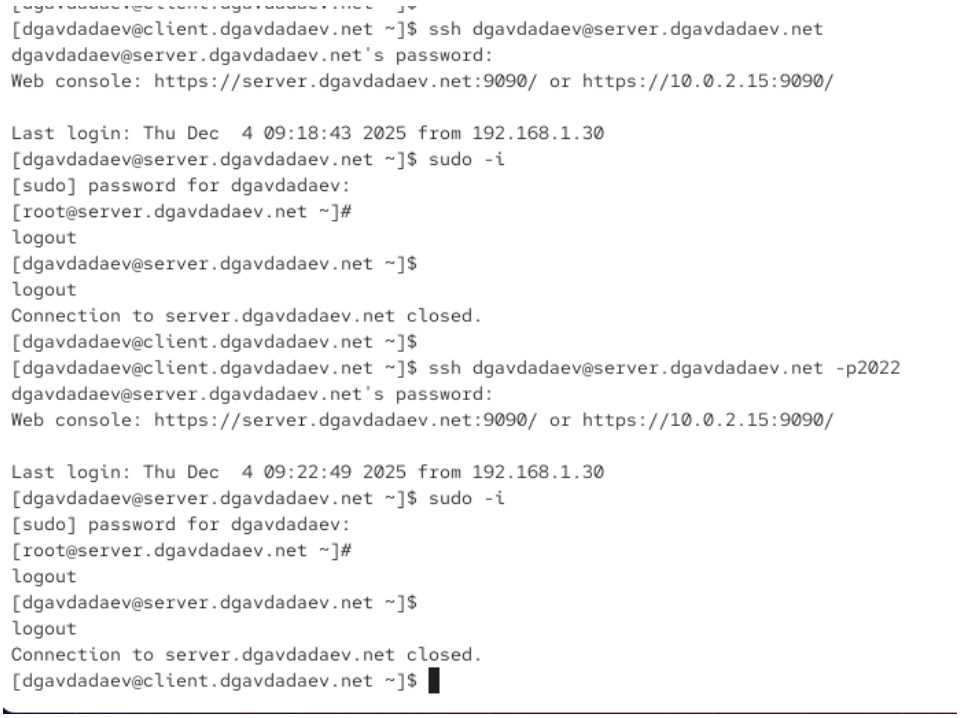

Рис. 2.12: Подключение по порту 2022

2.4 Настройка удалённого доступа по SSH с использованием ключей

2.4.1 Разрешение аутентификации по ключу

В конфигурационный файл SSH добавлен параметр:

```
PubkeyAuthentication yes
```

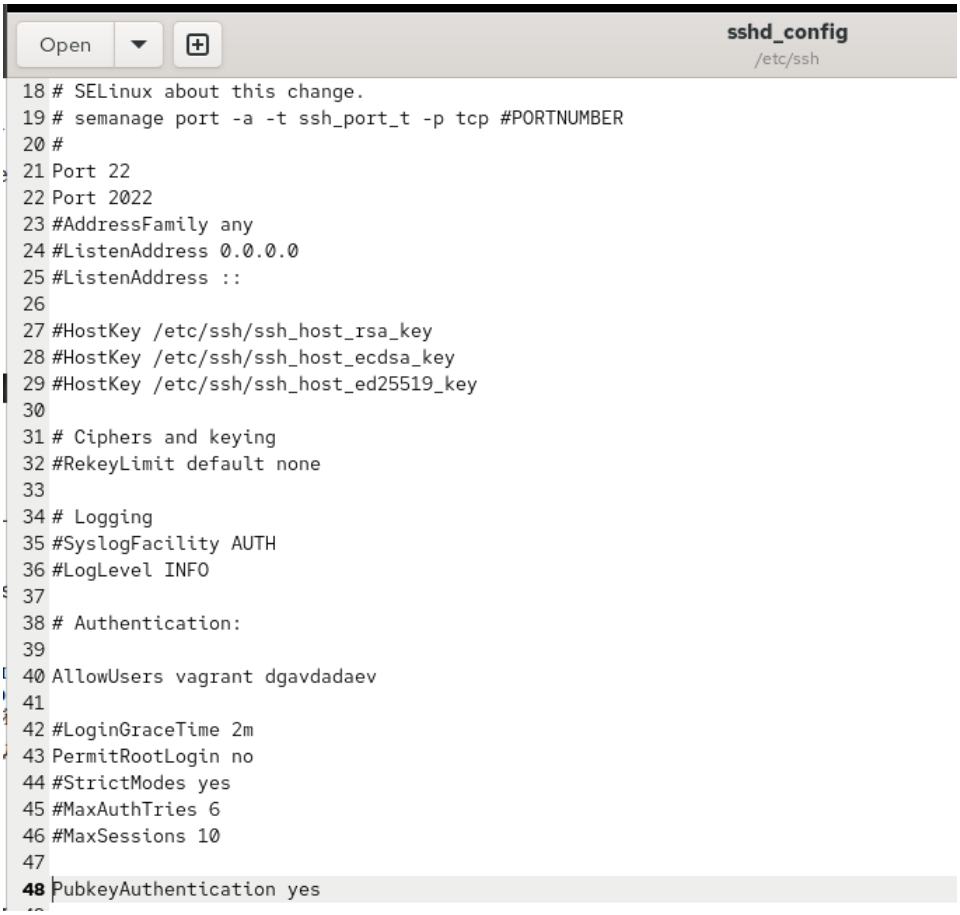

Рис. 2.13: Параметр PubkeyAuthentication

После внесения изменений sshd перезапущен.

2.4.2 Создание пары ключей на клиенте

Под пользователем *dgavdadaev* выполнено создание ключей:

```
ssh-keygen
```

При всех запросах подтверждений использованы настройки по умолчанию. В результате:

- закрытый ключ сохранён в `~/.ssh/id_rsa`
- открытый ключ — в `~/.ssh/id_rsa.pub`

2.4.3 Копирование ключа на сервер

Открытый ключ передан на сервер командой:

`ssh-copy-id [dgavdadaev@server.dgavdadaev.net](mailto:dgavdadaev@server.dgavdadaev.net`

После ввода пароля ключ был добавлен в `~/.ssh/authorized_keys` на сервере.

2.4.4 Подключение по SSH без пароля

Повторное подключение:

`ssh [dgavdadaev@server.dgavdadaev.net](mailto:dgavdadaev@server.dgavdadaev.net)`

Теперь вход выполняется без запроса пароля, что подтверждает успешную настройку SSH-доступа по ключу.

После завершения работы выполнен выход с сервера с использованием `Ctrl + d`.

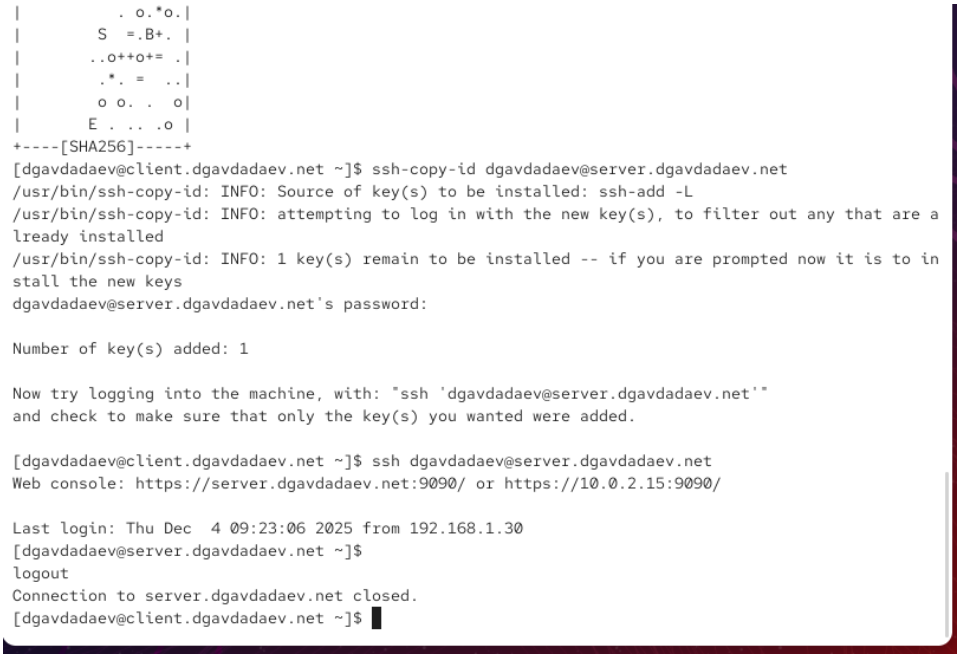

Рис. 2.14: Подключение по SSH без пароля

2.5 Организация туннелей SSH и перенаправление TCP-портов

2.5.1 Просмотр TCP-служб на клиенте

На клиенте был выполнен просмотр активных процессов, использующих TCP. На момент проверки никаких дополнительных служб не было запущено.

2.5.2 Создание SSH-туннеля для перенаправления порта

Было выполнено локальное перенаправление:

`8080 → server:80`

Команда открыла SSH-соединение и создала сокет, который принимает запросы на локальный порт 8080 и пересылает их на порт 80 сервера.

После создания туннеля повторная команда `lsof | grep TCP` показала новые записи:

- локальный процесс SSH слушает порт `localhost:webcache`
- активное TCP-соединение между клиентом и сервером
- перенаправленный сокет (LISTEN)

Рис. 2.15: Появление новых TCP соединений после создания туннеля

Это подтверждает, что порт-форвардинг активен.

2.5.3 Проверка работы перенаправленного порта

В браузере на клиенте открыт адрес:

http://localhost:8080

Страница успешно открылась, что означает корректную передачу трафика на веб-сервер удалённой машины.

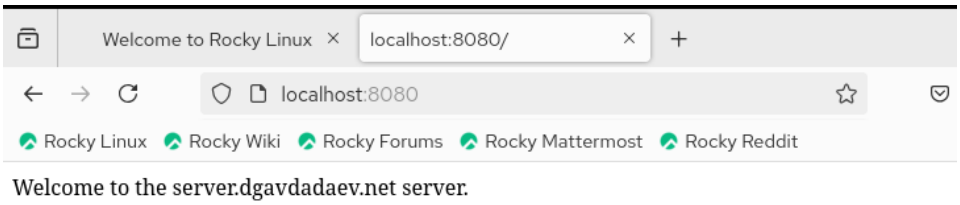

Рис. 2.16: Переход на localhost:8080 — отображение страницы сервера

2.6 Запуск консольных приложений через SSH

2.6.1 Просмотр имени узла сервера

Удалённая команда:

ssh dgavdadaev@server.dgavdadaev.net hostna

вернула имя сервера.

2.6.2 Просмотр файлов на сервере

Команда:

ssh dgavdadaev@server.dgavdadaev.net ls -
Al

отобразила содержимое домашнего каталога пользователя.

```
[dgavdadaev@client.dgavdadaev.net ~]$ ssh dgavdadaev@server.dgavdadaev.net hostname
server.dgavdadaev.net
[dgavdadaev@client.dgavdadaev.net ~]$ ssh dgavdadaev@server.dgavdadaev.net ls -Al
total 56
-rw-------.  1 dgavdadaev dgavdadaev  176 Dec  4 09:22 .bash_history
-rw-r--r--.  1 dgavdadaev dgavdadaev   18 Oct 29  2024 .bash_logout
-rw-r--r--.  1 dgavdadaev dgavdadaev  144 Oct 29  2024 .bash_profile
-rw-r--r--.  1 dgavdadaev dgavdadaev  549 Nov 25 07:12 .bashrc
drwx------. 11 dgavdadaev dgavdadaev 4096 Nov 25 09:00 .cache
drwx------. 10 dgavdadaev dgavdadaev 4096 Nov 25 07:14 .config
drwxr-xr-x.  2 dgavdadaev dgavdadaev    6 Nov 25 07:13 Desktop
drwxr-xr-x.  2 dgavdadaev dgavdadaev    6 Nov 25 07:13 Documents
drwxr-xr-x.  2 dgavdadaev dgavdadaev    6 Nov 25 07:13 Downloads
drwx------.  4 dgavdadaev dgavdadaev   32 Nov 25 07:13 .local
```

Рис. 2.17: Удалённый просмотр файлов

2.6.3 Просмотр почты пользователя

Команда:

`ssh [dgavdadaev@server.dgavdadaev.net](mailto:dgavdadaev@server.dgavdadaev.net) MAIL=~`

открыла консольный почтовый клиент и показала полученные сообщения.

```
[dgavdadaev@client.dgavdadaev.net ~]$
[dgavdadaev@client.dgavdadaev.net ~]$ ssh dgavdadaev@server.dgavdadaev.net MAIL=~/Maildir mail
s-nail version v14.9.24.  Type `?' for help
/home/dgavdadaev/Maildir: 3 messages 1 unread
    1 dgavdadaev            2025-11-30 11:04   18/668   "test1               "
    2 dgavdadaev@client.dg  2025-11-30 11:18   21/859   "LMTP test           "
▸U  3 dgavdadaev            2025-11-30 11:38   22/840   "test3               "
q
Held 3 messages in /home/dgavdadaev/Maildir
[dgavdadaev@client.dgavdadaev.net ~]$
```

Рис. 2.18: Просмотр почты через SSH

2.7 Запуск графических приложений по SSH (X11Forwarding)

2.7.1 Разрешение X11-переадресации на сервере

В файл `/etc/ssh/sshd_config` была добавлена строка:

```
X11Forwarding yes
```

Это разрешает пересылку X11-графики на клиентскую машину.

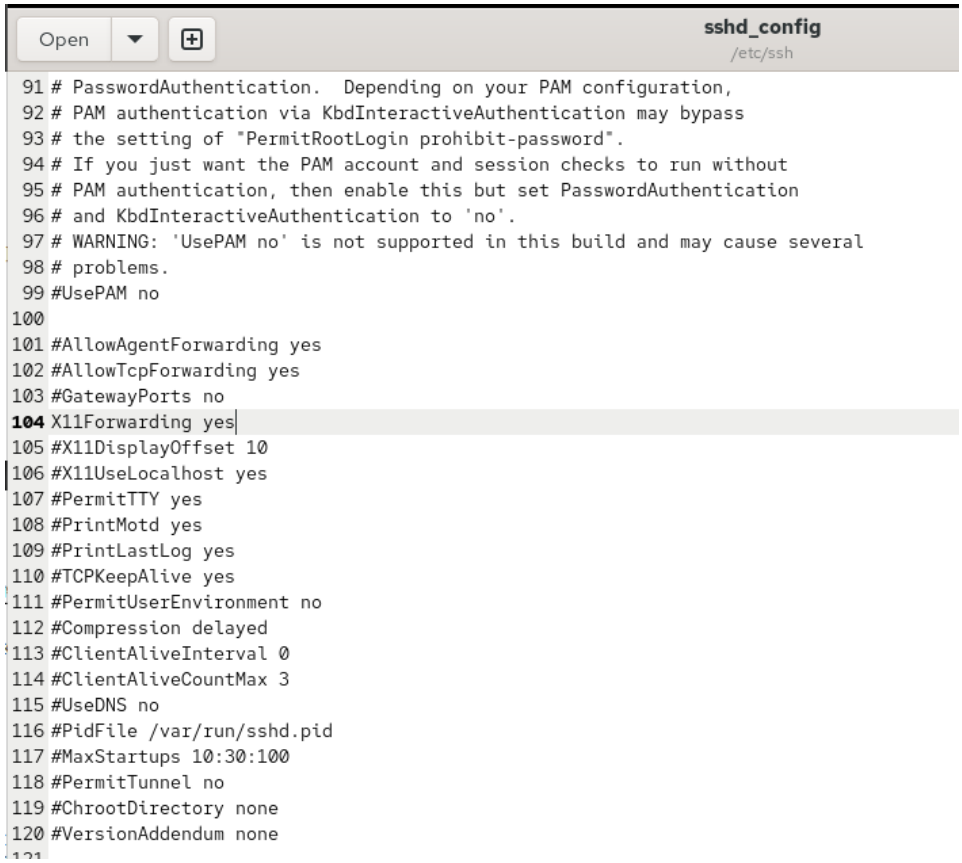

Рис. 2.19: Разрешение X11Forwarding

После перезапуска sshd была предпринята попытка запуска Firefox по SSH:

```
ssh -YC [dgavdadaev@server.dgavdadaev.net](mailto:dgavdadaev@server.dgavdadaev.net) fi
```

Однако запуск завершился ошибкой:

- отсутствует переменная DISPLAY на клиенте
- fake X-auth не может создать контекст
- пересылка X11 была отклонена сервером

Вывод:

- пересылка X11 настроена на сервере, но отсутствует полноценная X-среда в клиентской системе (например, Xorg, xauth или Wayland-X11 мост), поэтому графика не может быть отображена.

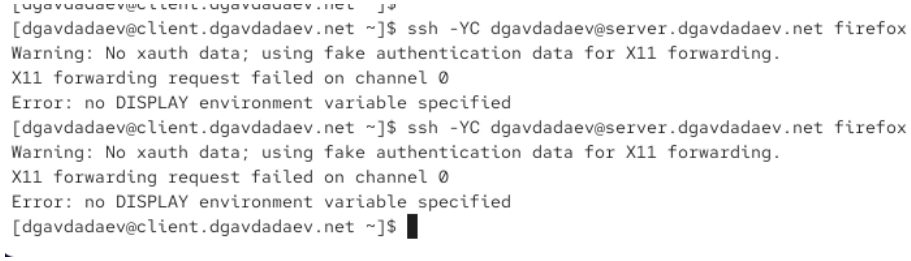

Рис. 2.20: Ошибка запуска графического приложения

2.8 Внесение изменений в настройки внутреннего окружения виртуальной машины

В каталоге `/vagrant/provision/server/` был создан каталог с конфигурацией SSH и подготовлен скрипт автоматизации.

2.8.1 Создание структуры каталогов и копирование конфигурации

```
mkdir -p /vagrant/provision/server/ssh/etc/ssh
cp -R /etc/ssh/sshd_config /vagrant/provision/server/ssh/etc/ssh/
```

2.8.2 Создание исполняемого файла `ssh.sh`

Скрипт был создан и сделан исполняемым.

Его содержимое:

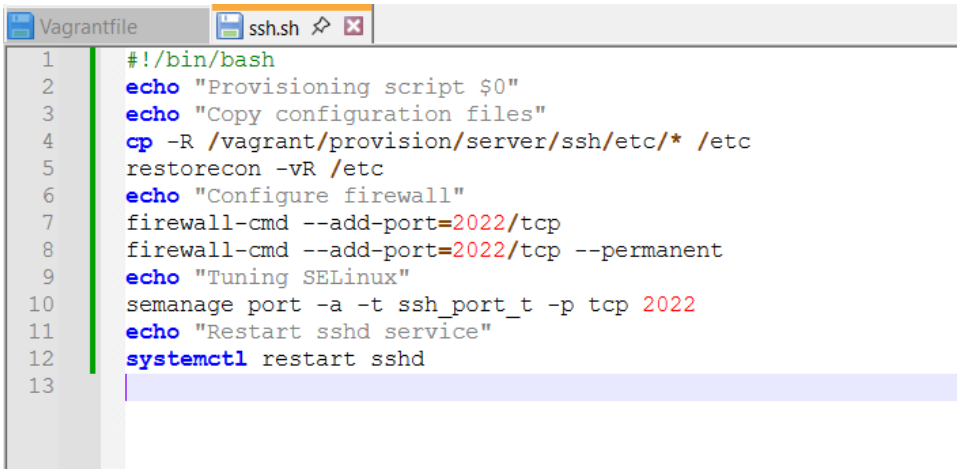

Рис. 2.21: Содержимое файла ssh.sh

Скрипт выполняет:

- копирование SSH-конфигурации внутрь Vagrant-окружения
- настройку firewall
- настройку SELinux
- перезапуск sshd

3 Итоги

3.1 Вывод

В ходе работы были настроены параметры SSH-доступа: запрещён вход root, разрешён вход выбранным пользователям, добавлены дополнительные порты и исправлены политики SELinux и firewall. Проверена работа SSH-туннелей, локальной переадресации портов, консольных команд и почтового клиента. Попытка запуска графических приложений выявила отсутствие X-среды на клиенте. Созданы и настроены файлы окружения Vagrant для автоматизации конфигурации SSH. Все задачи выполнены, доступ и инструменты работают корректно.

3.2 Контрольные вопросы

1. Вы хотите запретить удалённый доступ по SSH пользователю root и разрешить доступ пользователю alice. Как это сделать?

В конфигурации SSH (`/etc/ssh/sshd_config`) необходимо:

- запретить вход root, установив `PermitRootLogin no`;
- ограничить список пользователей строкой `AllowUsers alice`.

После изменения файла требуется перезапустить службу sshd. В результате вход root будет заблокирован, а пользователь alice сможет подключаться.

2. Как настроить удалённый доступ по SSH через несколько портов? Для чего это может потребоваться?

В файле `/etc/ssh/sshd_config` нужно указать несколько строк `Port`, например:

```
Port 22
Port 2022
```

После перезапуска sshd сервер будет принимать подключения на обоих портах. Это позволяет сохранить доступ к серверу в случае ошибки в настройках или при необходимости изменить стандартный порт для повышения безопасности.

3. Какие параметры используются для создания SSH-туннеля, когда ssh работает в фоне и не ожидает ввода команд?

Используются параметры:

- `-f` — перевести процесс в фоновый режим;
- `-N` — не выполнять удалённые команды;
- `-L` — настроить локальное перенаправление порта.

Такая комбинация позволяет создать устойчивый туннель без открытия удалённой оболочки.

4. Как настроить локальную переадресацию с локального порта 5555 на порт 80 сервера server2.example.com?

Используется локальный форвардинг:

```
ssh -fNL 5555:localhost:80 [user@server2.example.com](mailto:user@server2.example.com)
```

После выполнения команда создаёт туннель, через который обращения к `localhost:5555` пересылаются на порт 80 сервера.

5. Как настроить SELinux, чтобы позволить SSH связываться с портом 2022?

Необходимо добавить порт в тип ssh_port_t:

```
semanage port -a -t ssh_port_t -p tcp 2022
```

После добавления SELinux разрешит sshd использовать альтернативный порт.

6. Как настроить межсетевой экран на сервере, чтобы разрешить входящие подключения по SSH через порт 2022?

Команды для firewalld:

```
firewall-cmd --add-port=2022/tcp
firewall-cmd --add-port=2022/tcp --permanent
```

После перезапуска правил firewall новый порт становится доступным для входящих SSH-соединений.